

Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol 2, No.4, 2011

www.iiste.org

Design Approach to Implementation Of Arbitration Algorithm In Shared Bus Architectures (MPSoC)

D.Shanthi (Corresponding Author)

Research Scholar,HOD/ECE

Sri MuthuKumaran Institute of Technology

Mangadu,Chennai-69

E-Mail:dschelliah@yahoo.co.in

Dr.R.Amutha

Professor/ECE

SSN College of Engg.

Kalavakkam,Chennai

E-Mail:amuthar@ssn.edu

Abstract

The multiprocessor SoC designs have more than one processor and huge memory on the same chip. SoC consists of hardware cores and software cores ,multiple processors, embedded DRAM and connectors between cores .A wide range of MPSOC architectures have been developed over the past decade. This paper surveys the history of various On-Chip communication architectures present in the design of MPSoC. This acts as a primary factor of overall performance in complex SoC designs. Some of the various techniques that have driven the design of MpSoC has been discussed. Dynamically configurable communication architectures are found to improve the system performance. Currently On-chip interconnection networks are mostly implemented using shared buses which are the most common medium. The arbitration plays a crucial role in determining performance of bus-based system, as it assigns priorities, with which processor is granted the access to the shared communication resources. In the conventional arbitration algorithms there are some drawbacks such as bus starvation problem and low system performance. The bus should provide each component a flexible and utmost share of on-chip communication bandwidth and should improve the latency in access of the shared bus. The performance of SoC is improved using the probabilistic round robin algorithm with regard to the parameters, latency. Thus in this paper various issues related to bus arbitration related to design of MPSoC is analysed.

Keywords: MultiProcessor System-on Chip (MPSoC),Shared Bus, OnChip Network, Latency

1.1 Introduction

Shrinking process technologies and increasing design sizes have led to highly complex billion transistor integrated circuits (ICs).As a consequence, manufacturers are integrating increasing number of components on a chip. A heterogeneous SoC might include one or more programmable components such as general purpose processor cores, digital signal processor cores, or application-specific intellectual property cores (IPs) as well as an analog front end, On-chip memory, I/O devices and other application specific components. In other words, SoC is an IC that implements most or all the functions of a complete electronic system .

Modern System-on-Chip (SoC) architectures comprise several components such as master and slave modules... Masters are active modules that send read requests or data to memories. Typical masters are CPUs and hardware accelerators such as DMAs,hash generators ,or graphics engines. Slave modules are passive components that react on requests and store data or respond to master requests with appropriate data. Slave modules are typically memories, on-chip buses ,or simple register banks.Some of the bus architectures of

various MPSoC has been reviewed. Memories can be distinguished into on-chip and off-chip memories. On-Chip memories feature low latencies but small capacities whereas the off-chip memories exhibit high latency with high capacities. Figure 1 shows an SoC example.

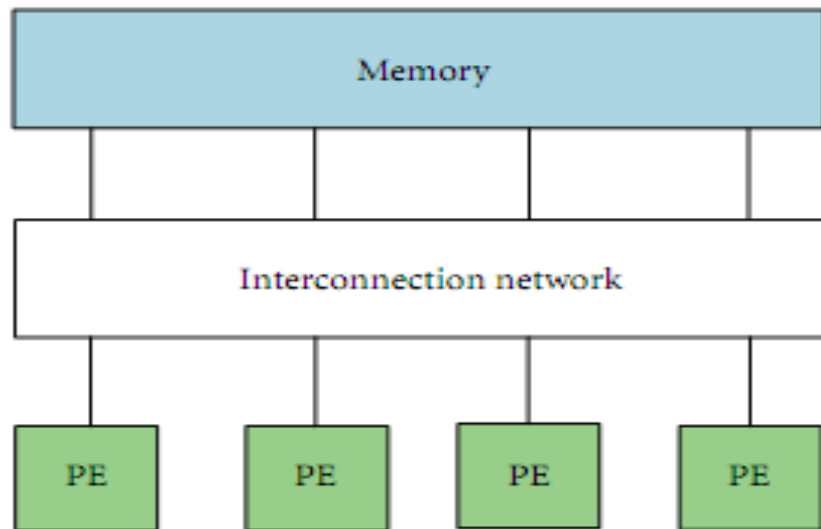


Fig 1:Shared Memory Multiprocessor System

Building an SoC requires a communication infrastructure that supports a large number of transaction masters and a large number of slaves, each of which can be arbitrated between the masters that needs to access that slave. Traditionally, multi-master buses like the Advanced High-performance Bus (AHB) have been used for the task. The bus arbiter resolves access conflicts by the masters to the single shared resource of the bus, which implicitly prevents multiple masters to access the same slave at the same time. A single shared bus does not allow any communication to happen in parallel between different masters and different slaves at the same time.

Concurrent requests at a slave are resolved using a simple arbiter. However simple arbitration lead to poor performance with regard to throput and latency of the overall system. Hence it is necessary to judiciously select a communication architecture that best suits or optimally suits the communication traffic generated for particular application. In addition to selecting communication architecture from a variety of alternatives, it is necessary to customize the selected architecture for the specific application or domain. Both these factors make it critical for a designer to be aware of, and to evaluate the trade-offs involved in the selection of an optimal system –level communication architecture.

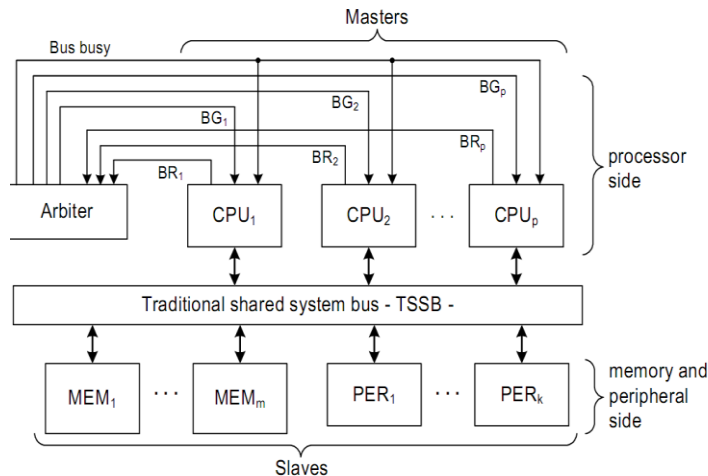


Fig:2 Traditional Shared Bus Topology

In this paper, we present a detailed survey and analysis of the performance of various commonly used SoC communication architectures, under several conditions. The architectures we consider in this paper includes Static Priority based shared system bus, hierarchical bus, TDMA based architecture and a ring based architecture. Efficient methodology has been developed to study the performance of various architectures.

In the next section, shared bus algorithms are discussed and arbitration process is analyzed. In Section II, various Shared bus-based architectures have been discussed. In Section III existing arbitration algorithm will be discussed. In Section IV, design issues of various types of shared bus architectures are analyzed and its features are have been discussed. Finally we conclude the review by discussing various factors related to communication architectures in SoC.

2.REVIEW ON SHARED BUS ARBITRATION ALGORITHMS

In this section, concepts and terminology associated with on-chip communication architectures has been introduced. Some popular communication architectures used in commercial SoC design is described. The communication architecture topology consists of a network of shared and dedicated communication channels, to which various SoC components are connected. These include (i) masters, which initiate a data transaction (e.g., CPUs, DSPs, DMA controllers etc.), and (ii) slaves, components that merely respond to transactions initiated by a master (e.g., on-chip memories). Fig (2). When the topology consists of multiple channels, bridges are used to interconnect the necessary channels. Since buses are often shared by several SoC masters, bus architectures require protocols to manage access to the bus ,which are implemented in (centralized or distributed) bus arbiters. Currently used communication architecture protocols includes round-robin, priority based and time division multiplexing .In addition to arbitration, the communication Protocol handles other communication functions like to limit the maximum number of bus cycles by setting maximum transfer length.

2.1. Static Fixed Priority:

It is a common scheduling mechanism (Bu-chung Lin et.al. 2007). In this scheme each master is assigned a fixed priority value. When several masters request simultaneously, the master with highest priority will be granted. This is achieved by employing a centralized arbiter. (Fig.2.).If masters with high priority requests frequently, it will lead to the starvation of the elements with lowest priority. But its advantage is its simple implementation and small area cost, flexibility and faster arbitration time. This protocol is used in shared bus communication architectures. (Fig. 2). This protocol is used by bus architectures like AMBA, Core Connect .

2.2. Time Division Multiple Access (TDMA):

Time division multiplexed scheduling divides execution time on the bus into time slots and allocates the time slots to adapters requesting the use of buses (Prakash Srinivasan et.al.2007). A request for use of the bus might require multiple slot times to perform all required transfers. If the master associated with current time slot has pending request ,the arbiter grants the transaction immediately and time wheel is rotated to next slot.

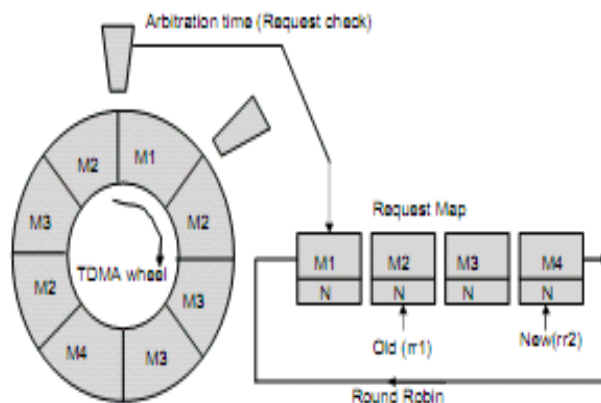


Fig 3:Schematic Diagram of TDMA Architecture

Advantage of this algorithm is that it is easy to implement. Disadvantage in this method is that it leads to the mistake of data transfer and poor response latency. However in this architecture, the components are provided access to communication channel in an interleaved manner, using two level arbitration protocols. To alleviate the problem of wasted slots, second level of arbitration is supported to permit the bus grant to other requesting masters. For e.g.. The current slot is reserved for M1, which has no pending request. As a result arbitration pointer is incremented from its current position to next pending request. (Fig 3). The major drawback is its poor bandwidth.

2.3.Round Robin Algorithm:

Round Robin algorithm can reallocate the available slots to other requesting master. It is a fair arbitration style when used with a limited transfer length. Whenever a turn ends, either unused or because of end of transfer or limited transfer length, the turn is passed to next component in order. Maximum access time and equal bandwidth can be achieved with limited transfer length. However it provides poor performance if requests are varied dynamically.

2.4. Lottery Bus Architecture:

In this protocol a centralized lottery manager accumulates request for ownership of shared communication resources from one or more masters ,each of which has assigned static or dynamic lottery tickets.Master owning the maximum number of tickets will be granted the access of bus.

2.5 Token passing Architecture:

This protocol is used in ring based architectures. A special data word, called token, circulates on the ring .An interface that receives a token is allowed to initiate a transaction. When the transaction is completed, the interface releases the token and sends it to the neighboring element. Ring based architectures have also been used in high speed ATM switches. The Fig. 4. Shows the model of ring based architectures with 8 components attached to the ring through ring interfaces.

The advantage of the ring based architecture is that the channel is connected to all the components, but is point-to-point and therefore can support higher clock rates than the previously described architectures. An important parameter is the maximum token holding time, which bounds the maximum number of words, a ring interface can send or receive each time it seizes the token.

2.6 Code Division Multiple Access (CDMA):

This protocol has been proposed for sharing on-chip communication channel. In a sharing medium it provides better resilience to noise/ interference and has an ability to support simultaneous transfer of data streams. But this protocol requires implementation of complex special direct sequence

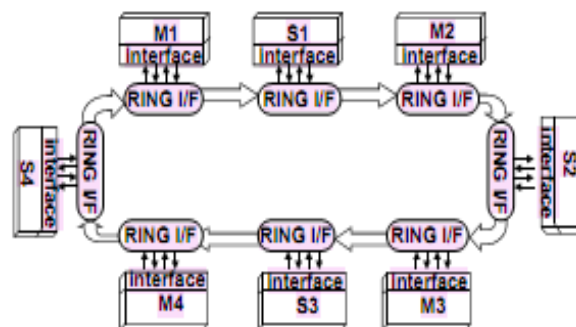


Fig 4: Ring based communication architecture

Spread spectrum coding schemes, and energy/battery inefficient systems such as pseudorandom code generators, modulation and demodulation circuits at the component bus interfaces and signaling (N. Shandhag 2004).

As a conclusion we can say that on-chip-bus-design and on-chip-core-based design methodologies are integration approaches that depend on standardized component or bus interfaces. They allow the integration of homogeneous IP components that follow these standards to be directly connected to each other, without requiring the development of complex wrappers. Let us note that on-chip buses rely on shared communication resources and on arbitration mechanism that is in charge of serializing bus access requests. This widely adopted solution unfortunately suffers from power and performance scalability limitations, and restricted sharing of resources between communicating entities. For bus networks, the bus is occupied by a single communication even if multiple communications could operate simultaneously on different portions on the bus. Therefore a lot of effort has been devoted to the development of advanced bus topologies (e.g. partial or full crossbar, bridged buses) and protocols for better support of route-ability, flexibility, reliability, and reconfigure-ability. Therefore, a systematic way of designing networks with possibly arbitrary topology is gaining the importance.

In the long run, a more aggressive approach is needed. For particular needs, the SoC may be built around a sophisticated and dedicated network-on-chip that may deliver very high performance for connecting a large number of components. It seems that this design paradigm shifts towards packetized on-chip communication based on micro-networks of interconnects or networks-on-chip (S.Hemachitra and P.T.Vanathi 2008).

3. SHARED BUS ARCHITECTURES

Various SoC buses are overviewed and its construction are discussed by Milica Mitić and Mile Stojčević (2006). Shared bus communication architectures like AMBA, WISHBONE, Core Connect, and PCI are most popular choices among the system designers due to their extensive features. AMBA arbiter design is simple enough to handle master and slave communication. Wishbone and Core connect arbiters and its design for communication between masters and slaves consume more area. The structure of the AMBA AHB is illustrated in the Fig.[5].

3.1 AMBA BUS:

AMBA is the most widely used bus communication in the emerging SoC applications. AMBA AHB developed by ARM consists of arbiter, masters and slaves. It allows arbiter to be designed to suit the application needs, the best. This specifies a hierarchy of bus types, tailored to differing priorities found across the interconnect structure of SoC designs. It minimizes silicon infrastructure required to support efficient on-chip and off-chip communication for both operation and manufacturing test.

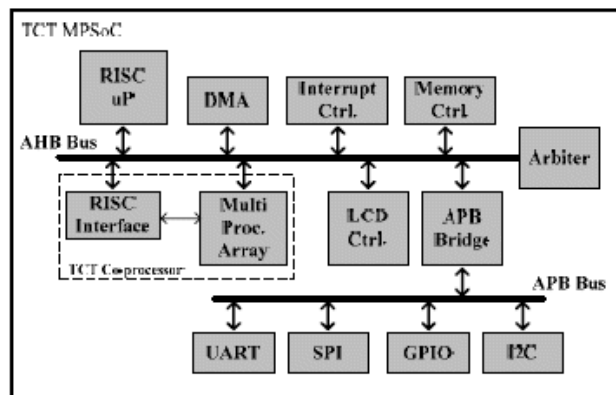


Fig 5: AMBA AHB Based SoC

3.2. Wish Bone BUS:

The Wishbone bus architecture is shown in Fig.[6].It shows a simple application of wishbone SoC Bus involving master slave communication..Wishbone uses master/slave architecture. Functional modules with master interfaces initiate data transactions to participating slave interfaces. There are four different types of interconnection in this architecture which includes Point-to-point, Data flow, Shared Bus and Crossbar switch. Arbiter selects the master that will own the slave, based on the arbitration technique. which can be chosen by the designer and implements it based on the application needs.

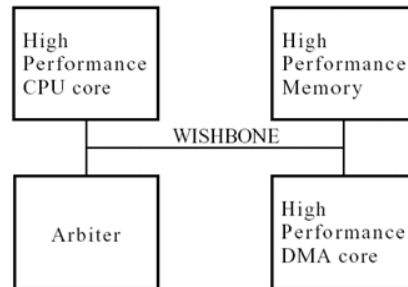


Fig 6:Wishbone SoC Bus

Wishbone doesn't have separate interfaces for low speed and high speed peripherals like AMBA. It appears to be the simplest of other buses reviewed.

3.3. Core Connect Bus:

Coreconnect is also more widely used bus communication architecture.Fig.7 shows a physical implementation of the arbiter. The arbitration is based on static priority with programmable priority fairness. Since the OPB supports multiple master devices, the address bus and data bus are implemented as a distributed multiplexer. This design will enable future peripherals to be added to the chip without changing the I/O on either the OPB arbiter or the other existing peripherals.

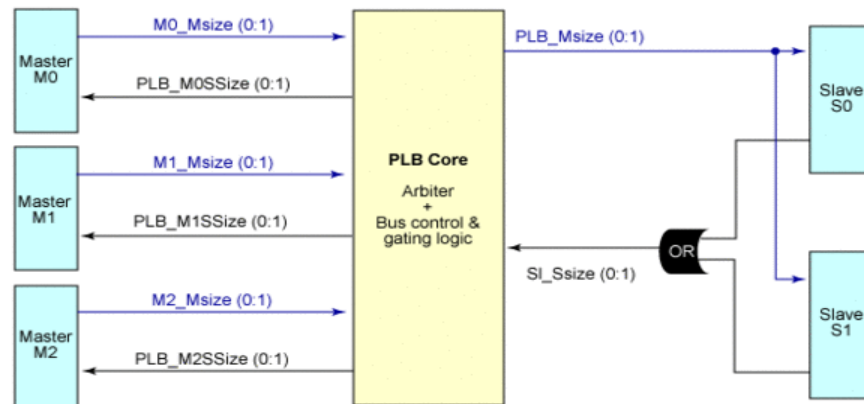


Fig 7: Coreconnect Bus Arbiter

4.EXISTING ON-CHIP BUS ARCHITECTURES

4.1. Lottery bus Communication Architecture

Lahiri et al (2001) presents a flexible and scalable algorithm for the multiprocessor SoC.The core of the LOTTERYBUS architecture is a probabilistic arbitration algorithm implemented in a centralized "lottery

manager” for each bus in the communication architecture. The architecture does not presume any fixed communication topology. Hence, the various SoC components may be interconnected by an arbitrary network of shared channels or a flat, system-wide bus.

The lottery manager accumulates requests for ownership of the bus from one or more masters, each of which is (statically or dynamically) assigned a number of “lottery tickets,” as shown in Fig. 8. The manager pseudo-randomly chooses one of the contending masters to be the winner of the lottery, favoring masters that have a larger number of tickets, and grants access to the chosen master for a certain number of bus cycles. Multiple word requests may be allowed to complete without incurring the overhead of a lottery drawing for each bus word. However, to prevent a master from monopolizing the bus, a maximum transfer size is used to limit the number of bus cycles for which the granted master can utilize the bus. Also, the architecture pipelines lottery manager operations with actual data transfers, to minimize idle bus cycles. The inputs to the

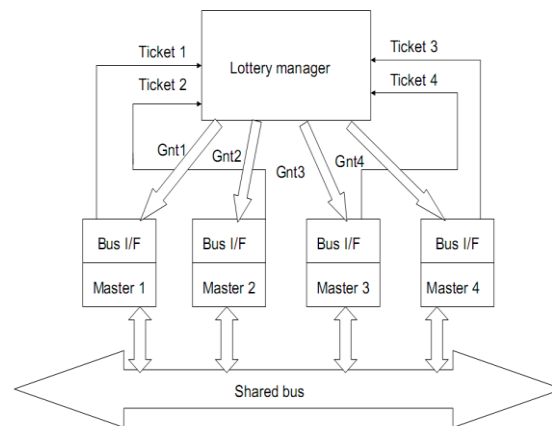


Fig 8: Lottery Arbiter for a Shared Bus

lottery manager are a set of requests (one per master) and the number of tickets held by each master. The output is a set of grant lines (again one per master) that indicate the number of words that the currently chosen master is allowed to transfer across the bus. The arbitration decision is based on a lottery. The lottery manager periodically (typically, once every bus cycle) polls the incoming request lines to see if there are any pending requests. If there is only one request, a trivial lottery results in granting the bus to the requesting master. If there are two or more pending requests, then the master to be granted access is chosen using the tickets t_0, t_1, t_2 and t_3 for the respective masters.

4.2 Dynamic Lottery bus architecture:

In this architecture, the inputs to the lottery manager consist of a set of request lines (r_0, r_1, r_2, r_3), and the number of tickets currently possessed by each corresponding master that are generated by ticket generator[8]. Therefore, under this architecture, not only can range of current tickets vary dynamically, it can take on any arbitrary value (unlike the static case, where it was fixed). Therefore at each lottery, the lottery manager needs to calculate for each

$$C_i, \text{ the partial sum } \sum_{j=1}^n r_j \cdot t_j.$$

This is implemented using a bit wise AND operation and adder tree of, as shown in Fig 4. The final result, $T = r_0 t_0 + r_1 t_1 + r_2 t_2 + r_3 t_3$, defines the range in which the random number must lie. A limitation of this implementation is that distribution of the resulting random number is not uniform. The rest of the architecture consists of comparison and grant hardware, and follows directly from the design of the static lottery manager.

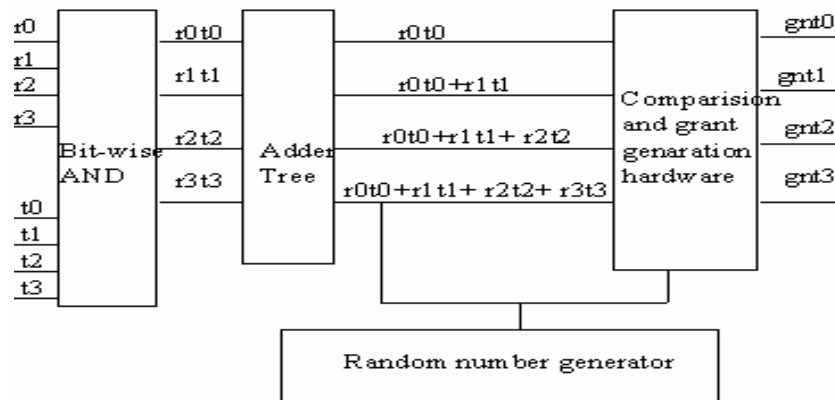


Fig 9: Lottery Arbiter for Dynamic varying Tickets

It is analyzed that advantage of Static Lottery Bus architecture is that all the masters requesting ,will gain control of the bus. Disadvantage of dynamic lottery bus architecture is that if pseudo random number is greater than the total ticket value none of the masters will get grant signal. Distribution of random number is not uniform for which strict uniform distribution should be desired.

4.3. ATM switch architecture:

In this arbitration algorithm, it accepts three parameters (Requests, Tickets, Adaptive signal) for the input of arbiter. Adaptive signal value is used as an additional input to improve the probability of the bus grant. This adaptive signal value is transmitted from the master that requires the bus grant more than another master because of the stressful traffic. Since we do not know which IP is used for the shared bus in advance of the SOC design, the adaptive signal can be fixed by the specific parameter. In this paper, the master counts the buffer position storing the ATM cell and if the data approaches to the limited amount, the adaptive signal is generated to improve the drawing probability].

The current pending request and ticket value is used to obtain the shared probability of C_i . In order to improve the probability of the master A_i values are obtained from the look up table and two of the master requests accomplish the bit-wise AND operation by the values. a_i is the additional ticket value to solve the problem of random value. [Fig.10].

We assume that the data approaches to the limited buffer capacity in C4 then master generates the adaptive signal. For the input adaptive signal, the MUX control signal is generated by the fixed value of the look up

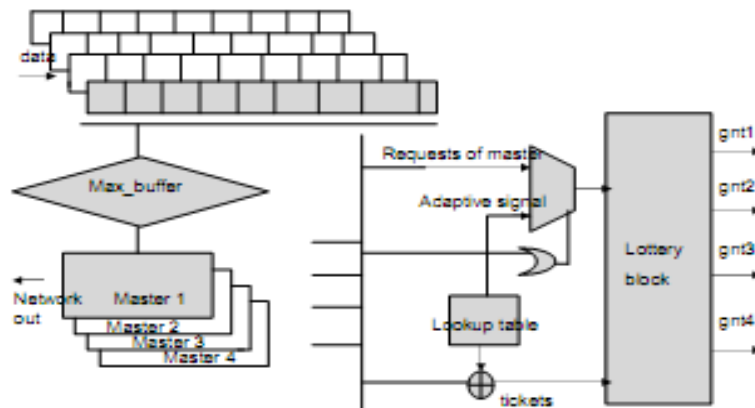


Fig 10:ATM Switch Architecture

table and the pending request value. In case C4 generates the adaptive signal and the adaptive request is 0001 from the lookup-table, the pending request value from the master is 1011 and the bitwise AND operation with 0001 from look-up table generates 0001. Therefore the total ticket value is

$$\sum_{j=1}^n r_j * t_j = 4$$

At the same time, the determined ticket value 0004 is generated and the existing ticket value 1034 assigned to the master operates with the adder to get the ticket value 1038. The adaptive ticket value is used to solve the problem that the characteristics of LFSR are disappeared.

If the pseudo random value is bigger than $\sum_{j=1}^n r_j * t_j = 4$, the control signal of MUX generates the enable

signal by the OR operation of the request bit from the master. The partial summation value of each master is obtained by the bit-wise AND operation between the request values and the ticket value. If the pseudo random value from

LFSR and the total ticket value generates modulo ($R, \sum_{j=1}^n r_j * t_j = 5$), C4 is assigned to use the bus, because the

pending request value is 0001. The advantage of ATM switch architecture is that the adaptive signal is used to solve the problem that the characteristics of LFSR are disappeared if pseudo random number is bigger than total ticket value.

5. REVIEW OF ARCHITECTURE DESIGN ISSUES:

The Fuzzy arbiter devised by Preeti et al. (2011) is found to be complex to implement and complexity increases with number of processors. It responds slowly since it requires many calculations and it is hard to implement.

Wei Zhang et al (2007) describes an MPSoC FPGA prototype based on hierarchy bus using 4 ARM processor cores. Satisfactory results have been achieved thru FPGA implementation and the platform works efficiently under higher workloads.

Yao et al (2006) proposed RB-Lottery algorithm which solves the starvation problem that exist in conventional algorithms and reduced average latency. The simulation shows, the algorithm has better performance of bandwidth requirements and has less average latency of bus requests than the lottery arbitration at the cost of increasing chip area and power consumption.

Ryu et al. (2001) have presented different MPSoC bus architectures and performance has been compared. It is concluded that bus architectures for a certain system must be determined by the type of application. The performance of these architectures is evaluated using applications from wireless communications, OFDM and MPEG 2 Decoder. It is found that among the five bus architectures Bi-FIFO and Cross Bar switch Bus Architecture perform the best for OFDM transmitter and MPEG 2 decoder respectively.

Sonntag and Helmut (2008) has devised weighted Round Robin algorithm to optimize the traffic characteristics in Multiprocessor architecture. It has been shown that WRR outperforms a round robin arbiter in throughput by 44% depending on the traffic patterns used. It is also proved latency for cache refill is also reduced by 34% using this arbiter.

Ari Kulmala et.al (2008) presents a thorough measurement of the effect of different arbitration algorithms on a real MPEG-4 implementation on FPGA. Various shared bus algorithms are compared. The measured quantities include video encoding performance, area usage and the effect of different maximum transfer

lengths. It is analyzed that at high utilization, Priority algorithm yield up to 60% better performance. At lower utilization, it is most preferable to use round-robin or combination of round-robin and priority with limited transfer length to avoid starvation.

The dynamic arbiter designed by Yi Xi Lu Lu et.al. (2006) can adjust the bandwidth proportion assigned to every processor automatically to avoid starvation problem in multiprocessor SoC technology. It is shown that the proposed arbiter can reduce 68% task execution time and bus request latency can be reduced to 78%. Better BW control can be provided in this priority arbiter than conventional arbiters.

Synthesis of Communication architectures containing multibus with bridges has been developed by A.Zitouni et.al.(2006) .This generates a hierarchical arbiter which is present inside the bridge .It allows to manage communication b/w components in same bus and also exchanges information between components belonging to different buses. A technique has been presented which minimizes the latency time between two successive occupations of the bus. The model developed seems to be compromise between multibus architectures and the NoC.The approach leads to the possibility of having simultaneous exchanges between components present on the same bus will be able to improve the performance of the target system.

Krishna Sekar and co. (2008) has designed FLEXIBUS, a new architecture that can efficiently adapt the logical connectivity of the communication architecture and the modules connected to it. It has been implemented as an extension of AMBA bus. They have applied in two SoC designs and performance has been analyzed. It is found that FLEXIBUS provides gains up to 34.55% compared to conventional architectures. Two hardware mechanisms are followed .Bridge by-pass and Component remapping which provides flexibility in adapting communication for various SoC components based on application specific environment Flexibus perform much better than static architecture and achieves a data rate improvement of 34.55% over single shared bus and 30.49% over multiple bus architecture.

6 SHORT-COMINGS ON EXISTING ARCHITECTURES

The limitations of the static priority based bus architecture and the two levels TDMA based architecture are discussed and the benefits of the LOTTERY bus communication architecture are demonstrated. The properties of the various arbitration styles have been discussed .Hence a flexible method of arbitration policy should be devised to suit the on chip communication architectures which overcomes some drawbacks faced. For e.g. consider the system shown in Fig .3. Static priority based arbiter is simpler in design and cost effective however there exists starvation of low priority components for the access of bus. Hence low priority components experience high latency. At times they may not have access for the bus, when a high priority component monopolizes the bus.

In TDMA/Round robin method there are defects such as bus starvation and low system performance due to distribution of slots for the master in a given bus cycle.It is concluded that the communication transaction latency is very sensitive to the time alignment of communication requests and the reservations of slots in the timing wheel .

Lottery Bus architecture improves the latency and provides low latency to high priority components. It is found that the latency of the highest priority component is lower than that of TDMA based architectures. The limitation of this method is that distribution of random number is not uniform. Hence another lottery method, ATM switch architecture overcomes the problem with regard to the characteristics of LFSR, if pseudo random number is greater than the total ticket value.

For the same number, multiple comparators may output a "1" (e.g. If $r_1 = 1$ and the generated random number is smaller than t_1 all the components will emit "1"), it is necessary to chose the first component. For example, for the request map 1011, assuming no scaling, if the generated random number is 5, only C4's associated comparator will output a "1". However ,if the generated random number is "1" ,then all the comparators will output "1" but the winner is C1.This requires for a standard priority selector circuit to ensure that ,for a given requests ,at the end of a lottery ,exactly only one grant line is inserted.

7 CONCLUSION

In this paper, we have discussed some of the issues related to the design of SoC with regard to the interprocessor communication .Various bus architectures and protocols have been reviewed. Currently on-

chip communication networks are mostly implemented using shared interconnects like buses. Shared bus communication architectures like AMBA, Wishbone, Core connect and PCI are most popular choices among the system designers due to their extensive features. AMBA is the most widely used bus communication in the emerging SoC applications. By power analysis between various architectures it is found that Core connect consumes more power due to presence of various gates in the interconnection [16]. Wishbone requires more area due to more interconnections in the architecture. Core connect consumes more latency due to gated signals from master to arbiter and from master to slave. Hence the designers should select the right arbitration technique to meet the requirements with improved performance for various shared bus architectures. Hence in the future research it is focused to design an arbiter that dynamically schedules the requests by various masters, occurring simultaneously and thus improving the performance of a multiprocessor with respect to latency and bandwidth.

References

- Sorren Sonntag and Helmut Reinig (2008), "An Efficient- Weighted-Round Robin algorithm for Multiprocessor Architectures," Proc. 41st Annual Simulation Symposium, IEEE Computer Society of India..
- IBM (2007), "IBM On-chip CoreConnect Bus Architecture." <http://www.chips.ibm.com/products/coreconnect/index.html>
- Bu-chung Lin, Geeng-Wei Lee, Juninn DarHuang and Jing-Yang Jou (2007), "A Precise Bandwidth Control Arbitration Algorithm for Hard Real Time SOC Buses", DAC 2007, pp 165-170
- Kanchan Warathe, Dinesh Padole and Dr.Preeti Bajaj (2009), "A Design Approach to AMBA Bus Architecture With Dynamic Lottery Arbiter", IEEE.
- Prakash Srinivasan, Ali Ahmadi, Ahmet, T Erdogan Tughrul Arslan (2007), "Power Evaluation of the Arbitration policy for different On-Chip Bus based SoC platform", IEEE SOC Conference, Taiwan, Volume, Issue, 26-29 Sept. 2007 pp:159 – 162.
- K. Lahiri, A. Raghunathan, Ganesh Lakshminarayana (2006), "The LOTTERYBUS on-chip Communication architecture", IEEE Trans. On VLSI System, June 2006.
- M.Kistler, M.Perrone and F.Petrini (2006), "Cell Multiprocessor Communication Network : Built For Speed", IEEE Trans on IEEE Micro, June 2006.
- K. Lahiri, A. Raghunathan, and S. Dey. (2001), "Evaluation of the Traffic Performance Characterization of System-on-Chip Communication Architectures" Proc. International Conference of VLSI Design, pp. 29-35, Jan. 2001
- N. Shandhag (2004), "Reliable and efficient system-on-chip design (2004)," IEEE Computer, vol. 37, no. 3, pp. 42–50, Mar. 2004.
- K. Lahiri, A. Raghunathan, and G.Lakshminarayan (2001), "LOTTERY BUS: New High Performance Communication Architecture for System-on-Chip Designs", Design Automation Conference, pp.15-20.
- Chang Hee Pyoun, et. All (2003), "The Efficient Bus Arbitration Scheme In Soc Environment", IEEE International Workshop System-on-chip.
- Milica Mitic and Mile Stojcev (2006), "An Overview of On-Chip Buses", Trans. On FACTA UNIVERSITATIS (NIS) SER.: ELEC. ENERG. vol. 19, no. 3, December 2006, pp: 405-428.
- W. Ho and T. Pinkston (2006), "A design methodology for efficient application-specific on-chip interconnects," IEEE Trans. On Parallel and Distributed Systems February, vol. 17, no. 2, pp. 174–190.
- J. L. A. et al (2006), "State-of-the-art soc communication architectures," in Embedded System Handbook, R. Zurawski, Ed. Boca Raton: CRC Taylor & Francis Group, pp: 20.1–20.2.
- S.Hemachitra and P.T.Vanathi (2008), "Design and Analysis of Dynamically Configurable Bus Arbiters for Socs", Trans. On ICGST, vol 8, Issue 1, Dec 2008.

W.Wolf,A.A.Jerraya,and G.Martin (2008) , “Multiprocessor System-on-chip (MPSoC) technology,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,vol .27,no. 10,pp. 1701-1713,

Dr.Preeti Bajaj and Dinesh Padole (2011) , “Arbitration schemes for multiprocessor Shared Bus,” New Trends and Developments in Automotive Engineering ,INTECH Publisher, Jan 2011.

Wei Zhang et al.(2007), “Design of Hierarchy Bus Based MPSoC on FPGA,” Proc. International Conference of Solid State and Integrated circuit Technology, April 2007,pp: 1966-1968.

Hsiao-Hui Yao,Chih-Peng Fan (2006), “Design and Implementation of A Lottery – based Bandwidth Guaranteed and Low Latency Arbiter for On-Chip Bus”, July 2006 White Papers at Silicon.com

M.Kistler,M.Perrone and F.Petrini (2006), “Cell Multiprocessor Communication Network :Built for speed”, Trans on IEEE MICRO, June 2006.

T.W.Ainsworth and T.M.Pinkstonv (2007), “ Characterising The Cell EIB On-Chip Network”,Trans on IEEE MICRO ,Sep-Oct 2007.

K.K.Ryu,E.Shin and V.J.Mooney (2001), “A Comparison of Five different Multiprocessor SoC Bus Architectures”,2001 IEEE.

(2001) Arm. amba multi-layer ahb overview. ARM. [Online]. Available: <http://www.arm.com>

(2003) Arm. amba axi protocol specifications. ARM. [Online]. Available: <http://www.arm.com>

(2002, Sept.) Wishbone system-on-chip (soc) interconnection architecture for portable IP cores, revision: B.3. [Online]. Available: <http://www.opencores.org/projects.cgi/web/wishbone/wishbone>

Ari Kulmala,Erno Salminen and Timo.D.Hamalainen (2008), “Distributed Bus Arbitration Algorithm Comparison on FPGA Based MPEG-4 Multiprocessor SoC,” IET Transactions on Computer and Digital Techniques, vol 2,issue 4 , pp: 314-325,July 2008.

Yi Xu Li Li Ming-lun Gao Bing Zhang Zhao-yu Jiang Gao-ming Du Wei Zhang (2006), “An Adaptive Dynamic Arbiter for Multiprocessor SoC,” Proc. International Conference of Solid State and Integrated circuit Technology, Oct 2006, Pg.1993-1996.

A.Zitouni,Sami Badrouchi and Rached Tourki (2006), “Communication Architecture Synthesis for Multi-bus SoC,” Trans. On Journal of Computer Science,2(1), pp: 63-71.

Krishna Sekar, Kanishka Lahiri ,Anand Raghuathan and Sujit Dey (2008), “Dynamically Configurable Bus Topologies for High-Performance On-Chip Communication,”IEEE Transactions On VLSI Systems,vol.16,no.10,OCT 2008,pp:.1413-1425.

C.P.Ravikumar (2004), “Multiprocessor Architectures for On-Chip Embedded Applications,”Proc.of 17 th International Conference on VLSI Design, 2004 IEEE Computer Society.

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:

<http://www.iiste.org>

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:**

<http://www.iiste.org/Journals/>

The IISTE editorial team promises to review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

